[Unit 10] Seminar 5: DR Solutions Design and Review

Any of the Businesses (AOB)

Why DR? CIA - the security triad

View DR (focus on system side, driven by business impact assessment) as a subset of BC (more general end-to-end business strategy)

- RTO - Recovery Time Objective: how quickly you can recover system when disaster strikes
- RPO - Recovery Point Objective: how much data you can afford to lose and how to recover data

DR Solutions Considerations

- On premise
    - Computer server, array of hard disk backups
    - Regulations or guidance: think along the lines of access control, how far does the DR have to be from the main system? Not in the same building, at least 21 miles apart able to operate independently in the event of an explosion
    - Not so much about NIST
- Cloud based
    - Downsides: not standardised, how you can migrate data onto cloud in case it doesn't work
    - Responsibilities - cloud, business, joint. (e.g. AWS)
    - Confidentiality, security

    - e.g. Switzerland has a requirement that all Swiss data must be held in Switzerland, same with UK, US…
    - Not enough to just say, I want my data centre to be in the UK, and perform backups - where were those backups going? How many copies are there, of those backups? There may be data centres in other countries and availability zones, regions. Not as easy to be able to predict where that data is actually being held.
- Hybrid - try to get the best of both worlds
- DRaaS - same concerns of where data is held, you get even less control of your data by outsourcing to a 3rd party

What makes a solution High Availability (HA)? RTO and RPO

Blue-Green deployment [DevOps]: multiple servers with load balancers, two identical systems in different locations. In most situations it is overkill (in terms of cost effectiveness) to have local HA, remote HA.

- Hot standby: Switch over at a moment's notice. Both running simultaneously, ready at all times.
- Regular testing and updating. Failover and redundancy. Deploying to each of them alternatively - update each system.
- Traffic manager - round robin system
- Delay in updates
- Active-passive (warm standby) vs active-active (hot standby)
    - Active-passive saves on running costs, but has a cold start problem (e.g. deployed but not running)

- Cold standby takes a longer time to spin up
- AWS reserved instances

DR Solution Backup

- What do I have to back up (e.g. VM images) vs what can I recreate (e.g. running data)?
- Proprietary solutions vs home-brew [open source]
    - Proprietary like ASR (Azure Site Recovery), NetApp example are costly where you contact the provider, you don't have control over the process
    - Home-brew like rsync - batch schedule, self managed
    - Automated vs manual
- Trade-off between utility vs cost vs safety
- Cloud storage (hot vs cold)
    - Tape backups - pseudo offline storage
- Off-site, off-cloud
- E.g. Network rail company stores off-site as it has petabytes of data, not cost effective to use cloud storage
- Small companies may not have the physical infrastructure set up, might use cloud services
- How to ensure you have multiple copies of data (cloud can be hacked/corrupted)

Shared Responsibility Model

Part A Reading: *Opara-Martins et al (2014), Morrow et al (2021)*

Main Vendor Lock-in Issues

- Migration and interoperability
    - Migration both ways - think Hotel California problem (no charges to move in data but charges apply for retrieving data out)
    - Interoperability with providers like AWS and Google, which have their own solutions - think about moving away from these solutions (move data, re-engineer system)
- Mitigation strategies
    - For systems like broadcasting announcement messages, might as well use proprietary systems to take advantage of the existing convenient infrastructure and since there is less emphasis on securing or migrating data.
    - Keep data out of cloud systems

Modern Cloud Security Concerns

- The cloud can be hacked, meaning they are as vulnerable as customers
- No control over your data (e.g. highly sensitive)
- User configurations (e.g. bucket storage)
- Limit access to certain endpoints (e.g. not default to port 22 to ssh from anywhere)
- Do not commit and leave files on GitHub with passwords and secrets

References

Opara-Martins, J., Sahandi, R., & Tian, F. (2014) Critical review of vendor lock-in and its impact on adoption of cloud computing. *In International Conference on Information Society* (i-Society 2014) (pp. 92-97). IEEE.

Morrow, T., LaPiana, V., Faatz, D., Hueca, A. & Richmond, N. (2021) *Cloud Security Best Practices Derived from Mission Thread Analysis.* Carnegie-Mellon Univ Pittsburgh PA.